

LISTING OF CLAIMS

1. (currently amended) A system for automated testing of software, the system comprising a system server component comprising,
 - a test bucket for storing sets of test data,
 - a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,
 - a resource process and resource pool for managing system resource data to indicate resources available for software testing on a set of client computer systems,
 - a job execution process for creating test execution script data based on the test data identified in a test request and the available resources,
 - wherein the job execution process receives the test request from the job receiver process and receives input from the resource process indicating resources available for software testing,
 - dynamically creates the test execution script based upon the resource pool indicating the availability of resources required for the execution of the test on one or more of the set of client computer systems, and
 - initiates testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and
 - the system server component further comprising a means for accepting and storing test results from the set of client computer systems.
2. (original) The system of claim 1 further comprising a client process component, the client process component being

executable on one or more of the set of client computer systems and comprising

a listener process for accepting test execution script data from the system server component,

a test execution process for carrying out the testing specified by test execution script data provided by the listener process, and for generating a test report and for communicating the test report to the system server component.

3. (canceled)

4. (canceled)

5. (original) The system of claim 1, in which the system server further comprises an active job queue and a dispatcher process,

job receiver process placing test requests on the active job queue upon receipt,

the dispatcher process determining when a subject test request on the active job queue is matched by available system resources as indicated by the resource pool and providing the subject test request to the job execution process.

6. (original) The system of claim 5 further comprising a complete job queue for receiving test requests from the job execution process upon the completion of the testing defined by the test request.

7. (original) The system of claim 1 in which the system server component further comprises a database for the storage of test results received by the job execution process.

8. (original) The system of claim 1 in which the system server comprises TCP/IP sockets for accepting test requests and communicating with the set of client systems.
9. (original) The system of claim 2 in which the listener process generates a test script file from the test script data received from the system server component and which test script file used by the test execution process to define the testing carried out by the client process.
- D
10. (original) The system of claim 2 in which the test execution process generates a report file which is returned to the system server on completion of testing.
11. (original) The system of claim 2 in which the client process further comprises a client configuration file for selectively defined pre-testing and post-testing configuration of the client system computer on which the client process is executing.
12. (original) The system of claim 2 in which the client process further comprises a control process for receiving control queries and commands from the system server component and for responding to the control queries and commands and in which the job execution process in the system server component further comprises means for generating control queries and commands and for receiving responses to the control queries and commands.
13. (original) The system of claim 12 in which the control queries and commands comprise a refresh command, the system further comprising an automated machine refresh subsystem responsive to the refresh command.

14. (original) The system of claim 13 in which the automated machine refresh subsystem is for DOS- based client system computers running a non-DOS operating system for software testing, the automated machine refresh subsystem comprising
a stored machine image,
a refresh script for modification of the boot.ini and autoexec.bat files on client system computers,
the modified autoexec.bat file being configured to modify the boot.ini file and execute drive image software for loading the stored machine image and for rebooting the system into the non-DOS operating system for software testing.

15. (currently amended) A computer program product for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium for software testing in distributed systems, said computer program product comprising:

computer readable program code means for causing a computer to define and manage a test bucket for storing sets of test data,

computer readable program code means for causing a computer to execute a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

computer readable program code means for causing a computer to execute a resource process for managing system a resource pool to indicate resources available for software testing on a set of client computer systems,

computer readable program code means for causing a computer to execute a job execution process for creating test execution script data based on the test data identified in a test request and the available resources,

wherein the job execution process receives the test request from the job receiver process and an indication of the available resources from the resource process,

dynamically creates the test execution script data based the resource pool indicating the availability of resources required for the execution of the test on one or more of the set of client computer systems, and

initiates testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

computer readable program code means for causing a computer to accept and store test results from the set of client computer systems.

16. (original) The computer program product of claim 15 further comprising

computer readable program code means for causing a computer to execute a client process component, the client process component being executable on one or more of the set of client computer systems and comprising

computer readable program code means for causing a computer to execute a listener process for accepting test execution script data from the system server component,

computer readable program code means for causing a computer to execute a test execution process for carrying out the testing specified by test execution script data provided by the listener process, and for generating a test report and for communicating the test report to the system server component.

17. (currently amended) A computer program product tangibly embodying a program of instructions executable by a computer for implementing a system for automated testing of software, the system comprising a system server component comprising,

a test bucket for storing sets of test data,
a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,
a resource process and resource pool for managing system resource data to indicate resources available for software testing on a set of client computer systems,
a job execution process for creating test execution script data based on the test data identified in a test request and the available resources by the steps of: [,by]
receiving the test request from the job receiver process and an indication of available resources from the resource process,
dynamically creating test execution script data based upon the resource pool indicating the availability of resources required for the execution of the test on one or more of the set of client computer systems, and
initiating testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and
the system server component further comprising a means for accepting and storing test results from the set of client computer systems.

18. (original) The computer program product of claim 17, the system for automated testing of software further comprising a client process component, the client process component being executable on one or more of the set of client computer systems and comprising

a listener process for accepting test execution script data from the system server component,
a test execution process for carrying out the testing specified by test execution script data provided by the listener

CA919990016

process, and for generating a test report and for communicating the test report to the system server component.

19. (currently amended) A method for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium for software testing in distributed systems, said method comprising the steps at said computer of:

defining and managing a test bucket for storing sets of test data,

executing a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

executing a resource process for managing ~~system~~ a system resource pool to indicate resources available for software testing on a set of client computer systems,

executing a job execution process for creating test execution script data based on the test data identified in a test request and the available resources, by performing the steps of:

receiving the test request from the job receiver process at the job execution process and resource availability from the resource process,

dynamically creating a test execution script indicating the availability of resources required for the execution of the test on one or more of the set of client computer systems, and

initiating testing at said job execution process by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

accepting and storing test results from the set of client computer systems.

20. (previously presented) The system of claim 1 further comprising a web servlet component providing a graphical user interface for use by the user in defining a test request.

21. (previously presented) The system of claim 2 further comprising a web servlet component providing a graphical user interface for use by the user in defining a test request.

22. (previously presented) The system of claim 1 further comprising a parser component for parsing ASCII format test requests defined by the user.

23. (previously presented) The system of claim 2 further comprising a parser component for parsing ASCII format test requests defined by the user.

24. (previously presented) The system of claim 3 further comprising a parser component for parsing ASCII format test requests defined by the user.